Bug Tracker

Introduction

A bug tracker is a software tool designed to help software development teams track and manage the reported issues or bugs in their projects. It is an essential component of the software engineering process, as it facilitates efficient collaboration, bug resolution, and project management.

Bug tracker software provides a centralized platform for recording, organizing, and tracking the lifecycle of bugs or issues reported by testers, users, or team members. It allows developers, testers, and other stakeholders to communicate effectively, prioritize tasks, and ensure timely bug fixes.

Part1:

Functional Requirements:

1. Users can report bugs by filling out a form that includes a title, description, and steps to reproduce.
2. Users should be able to assign bugs to specific team members and track the progress of each bug.
3. Users should be able to search and filter through reported bugs based on various criteria such as priority, status, and severity.
4. Users can comment on and discuss bugs with other team members.
5. The bug tracker should provide notifications to relevant stakeholders about changes to bug status, assignment, or comments.
6. Users should login or sign in into account in order to report bugs.
7. Bug tracker should send an email to the user when logging in to account or creating a new account.
8. The system should allow users to attach files, screenshots, and other relevant material to bug reports.
9. The system should provide a history of all changes made to each bug reports including the user information who made the change and the date and time it was made.
10. The bug tracker should allow users to schedule bug fixes and set deadlines for resolution.
11. The bug tracker should provide a dashboard or other visualization tools to help users track the status of bugs and other project data.

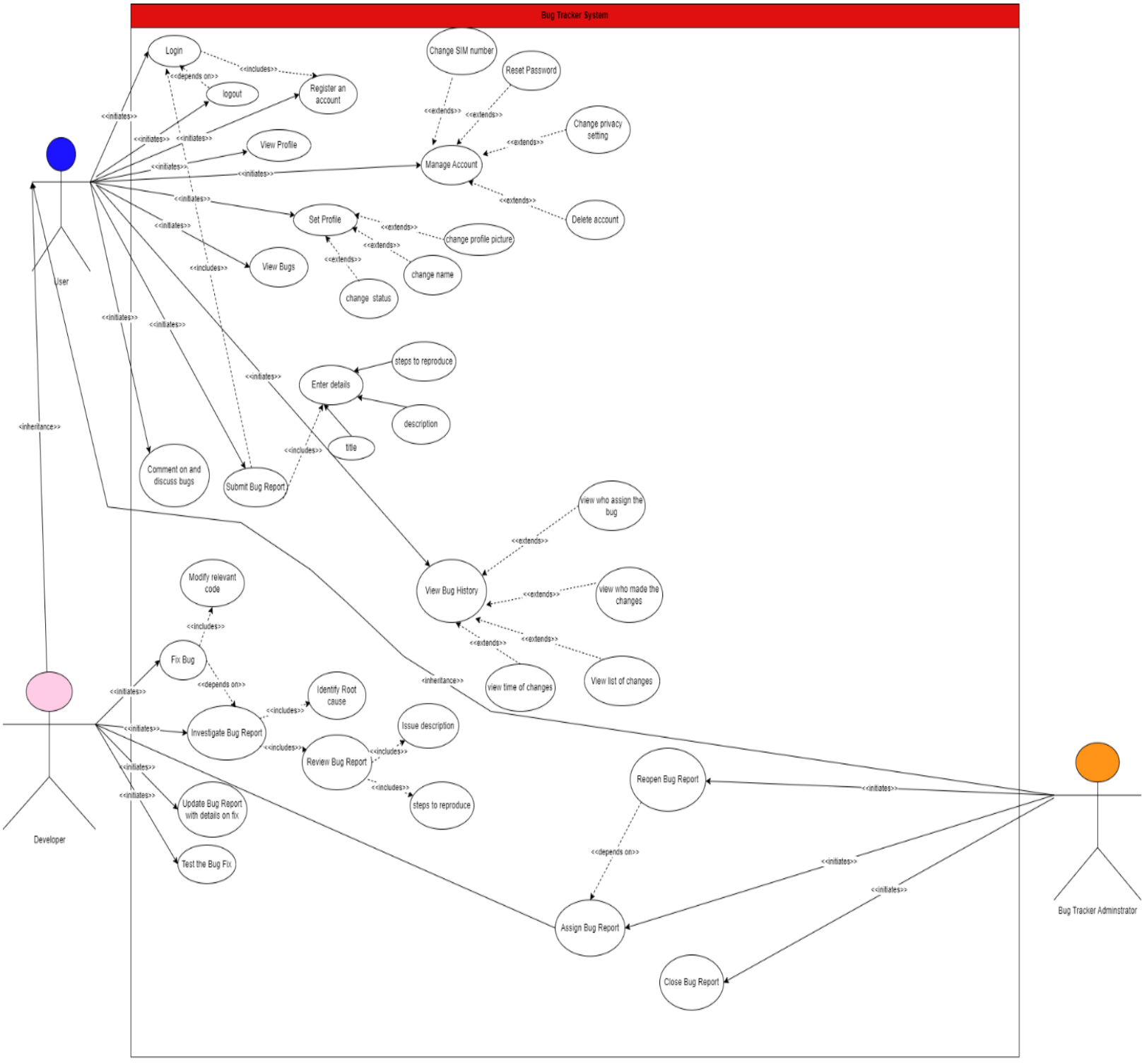
Non-Functional Requirements

1. The bug tracker should have a fast response time, allowing users to quickly navigate the application.
2. The bug tracker should be reliable and provide accurate data at all times
3. The bug tracker should comply with relevant laws and regulations, such as data privacy and security standards.
4. The bug tracker should allow users to export and download data in multiple formats: ensuring compatibility with other tools and services.
5. The bug tracker should provide a mechanism for users to give feedback or suggestions for improving the application.
6. The bug tracker should be designed with a minimalist aesthetic, reducing visual clutter and making it easier to focus on important information.
7. The bug tracker should be available 24/7 and have a minimum uptime of 99.9%
8. The bug tracker should be secure and protect user data from unauthorized access or manipulation.
9. The bug tracker should have a responsive and user-friendly interface that can be used on various devices and screen sizes.
10. The bug tracker should be customizable and able to adapt to the unique needs of the project team.
11. The bug tracker should be scalable and able to handle an increasing number of users and bugs as the project grows.
12. The bug tracker should be able to integrate with other project management tools and services.
13. The bug tracker should be compatible with different operating systems and browsers, ensuring broad accessibility.
14. The bug tracker should be easy to maintain and update, minimizing downtime and disruption to users.

Part2:

Task#	Task	Duration (days)	Dependency	Number of Software Engineers	Number of Programmers
A	Problem Analysis	2	-	2	-
B	Requirements Analysis	4	A	3	4
C	Logical Design	3	A	4	5
D	Decision Analysis	6	B	2	3
E	Physical Design	5	C	2	2
F	Constructing and Testing	6	D	1	4
G	Implementation And Delivery	8	F,E	1	4

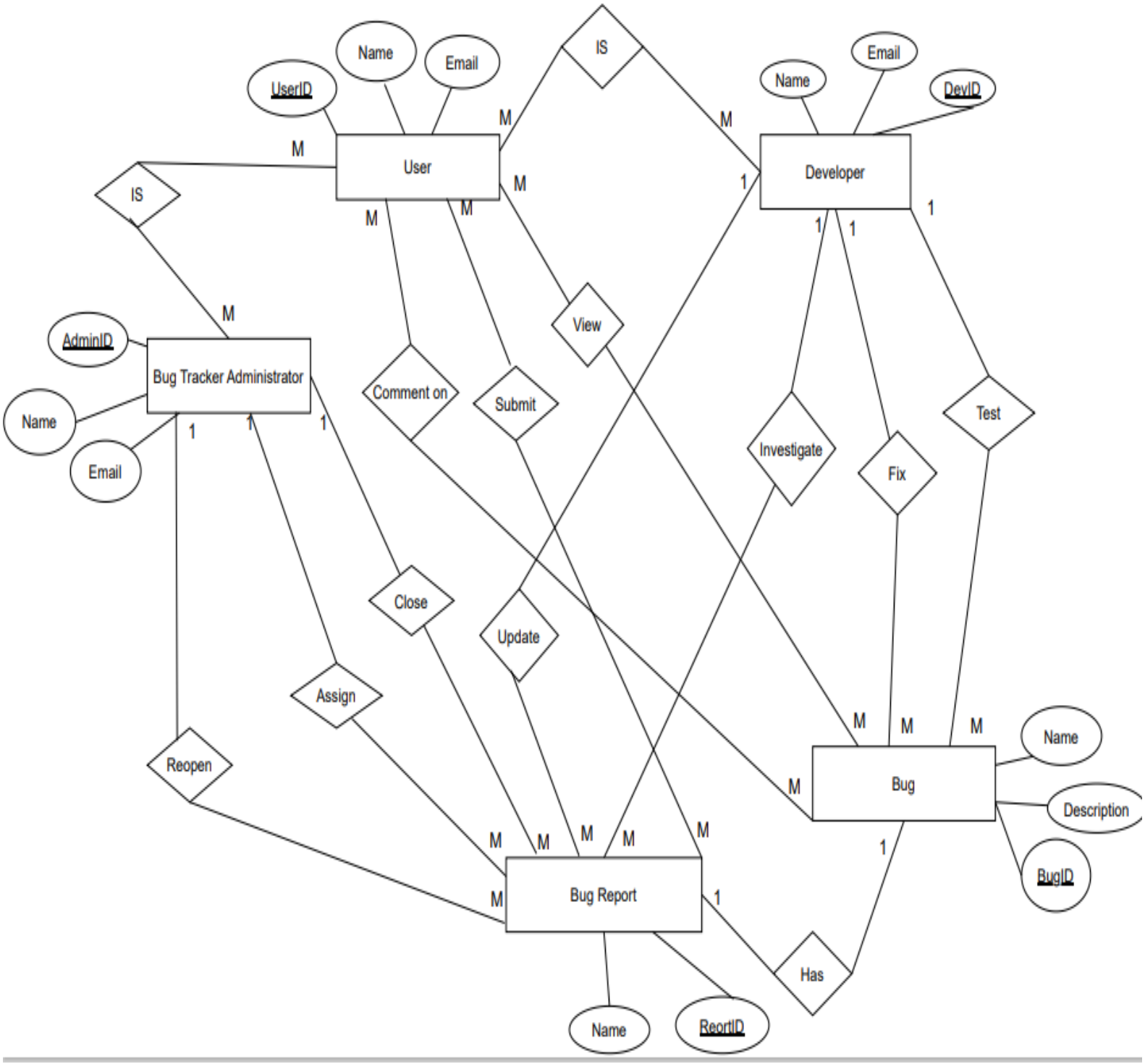
UseCase:


Bug Tracker System
Login
Logout
Register an account
Change SIM number
Reset Password
Change privacy setting
Manage Account
Delete account
View Profile
Set Profile
change profile picture
change name
change status
View Bugs
Comment on and discuss bugs
Submit Bug Report
Enter details
steps to reproduce
description
title
View Bug History
view who assign the bug
view who made the changes
view time of changes
View list of changes
Modify relevant code
Fix Bug
Identify Root cause
Investigate Bug Report
Issue description
Review Bug Report
steps to reproduce
Update Bug Report with details on fix
Test the Bug Fix
Reopen Bug Report
Assign Bug Report
Close Bug Report
User
Developer
Bug Tracker Adminstrator
<<initiates>>
<<includes>>
<<depends on>>
<<extends>>
<<inheritance>>

Use-Case Narrative

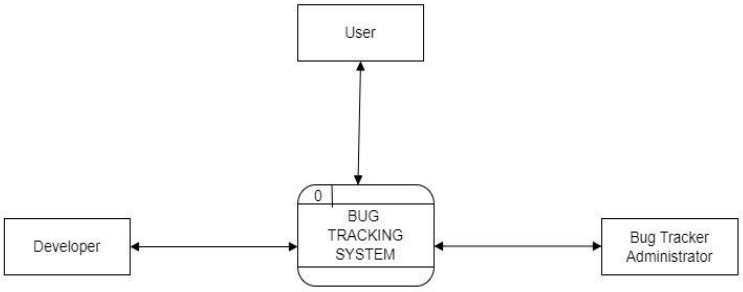
Use-Case Name:	Assign Bug Report	
Use-Case ID:	BT-002	
Priority:	High	
Primary Actor:	Bug Tracker Administrator	
Description:	This use case describes the event that the administrator is Assigning a new bug report to a specific developer.	
Precondition:	A bug report must exist in the system that needs to be assigned to a developer or team.	
Typical Course of Events:	**Actor Action**	**System Response**
	The Bug Tracker Administrator selects the "Assign Bug Report" option. The Bug Tracker Administrator selects the developer responsible for investigating and fixing the bug report.	The system checks if this developer is available. If the selected developer is available, the system notifies him with a new report that is responsible to fix. If he is not available, the system notifies the administrator to select another developer. The bug report is assigned to the selected developer or team. The bug tracker software sends a notification to the assigned developer.
Conclusion:	The bug report is assigned to the selected developer, and the administrator receives a notification showing that the developer receives his report successfully.	

ER DIAGRAM:


Name
Email
UserID
IS
Name
Email
DevID
User
Developer
IS
View
AdminID
Bug Tracker Administrator
Name
Email
Comment on
Submit
Investigate
Fix
Test
Close
Update
Assign
Reopen
Name
Bug
Description
BugID
Bug Report
Has
Name
ReortID
M
1

DFD DIAGRAM:

Context Diagram



DFD